

Image Classification and Retrieval Using Elastic Shape Metrics

Shantanu H. Joshi
Department of Electrical Engineering
Florida State University
Tallahassee, FL-32310
joshi@eng.fsu.edu

Anuj Srivastava
Department of Statistics
Florida State University
Tallahassee, FL-32306
anuj@stat.fsu.edu

Abstract—This paper presents a shape-based approach for automatic classification and retrieval of imaged objects. The shape-distance used in clustering is an intrinsic elastic metric on a nonlinear, infinite-dimensional shape space, obtained using geodesic lengths defined on the manifold. This analysis is landmark free, does not require embedding shapes in \mathbb{R}^2 , and uses ODEs for flows (as opposed to PDEs). Clustering is performed in a hierarchical fashion. At any level of the hierarchy, clusters are generated using a minimum dispersion criterion and a MCMC-type search algorithm is employed to ensure near-optimal configurations. The Hierarchical clustering potentially forms an efficient ($\mathcal{O}(\log(n))$ searches) tool for retrieval from large shape databases. Examples are presented for demonstrating these tools using shapes from the ETH-80 shape database.

Keywords: *shape clustering, shape classification, image retrieval*

I. INTRODUCTION

Unsupervised learning of visual object features is an important task in machine vision applications such as medical imaging, automatic surveillance, biometrics, and military target recognition. The imaged objects can be characterized in many ways: according to their colors, textures, shapes, movements, and locations. Of late, shape has been used as an important discriminant for identification and recognition of objects from images. Indeed, it is a desirable goal for an intelligent system to have automated tools for classifying and clustering objects according to the shapes of their boundaries.

A. Past Shape-based Image Retrieval

In general, there have been numerous approaches for including shapes in conjunction with color, intensity and textures for image indexing and retrieval. Many techniques, including Fourier descriptors [18], [17], Wavelet descriptors [20], chain codes, polygonal approximations [19], and moment descriptors [21] have been proposed and used in

various applications. Cortelazzo et al. [16] use chain codes for trademark image shape descriptions and string matching techniques. Jain and Vailaya [12] propose a representation scheme based on histograms of edge directions of shapes. A different approach by Mokhtarian et al. [14] uses curvature scale space methods for robust image retrieval from the Surrey fish dataset [13]. A majority of these methods have focused on the limited goal of fast shape matching and retrieval from large databases. Simple metrics using either Fourier or moment descriptors, or scale-space shape representations, may prove sufficient for retrieving shapes from a database. However they lack the tools and the framework for more advanced analysis, especially if one requires building probability models using the retrieved results.

B. Past Shape Analysis Methods

To address the above difficulties, and seek a a full statistical framework, Klassen, Srivastava et al. [2] adopt a geometric approach to parameterize curves by their arc lengths, and use their angle functions to represent and analyze shapes. Using the representations and metrics described in [2], Srivastava et al. [5] describe techniques for clustering, learning, and testing of planar shapes. One major limitation of this approach is that all curves are parameterized by arc length, and the resulting transformations from one shape into another are restricted to *bending only*. Local stretching or shrinking of shapes is not allowed. Mio and Srivastava [3] resolve this issue by introducing a representation that allows both bending and stretching of curves to compare and match shapes. It has been demonstrated in [3], that geodesics resulting from this approach seem more natural as interesting features, such as corners, are better preserved, thus leading to improved metrics in the shape space. We use the approach presented in [3] to represent and analyze shapes of closed curves. The basic idea is to represent



Fig. 1. Example of a geodesic between a pair of shapes.

these curves as parameterized functions, not necessarily arc-length, with appropriate constraints, and define a non-linear manifold \mathcal{C} of closed curves. To remove similarity transformations, one forms a quotient space $\mathcal{S} = \mathcal{C}/S$, where S is the space of similarity transformations. Shapes of closed curves are analyzed as elements of \mathcal{S} . The following section describes the shape representation scheme and briefly explains the construction of geodesics between any two given shapes on \mathcal{S} .

C. Elastic Shape Representation Scheme

Let β be a parameterized curve of interest, of length l , and $\alpha = 2\pi\beta/l$ be its re-scaled version. We will assume $\alpha: [0, 2\pi] \rightarrow \mathbb{R}^2$ to be a smooth, non-singular, orientation-preserving, parametric curve in the sense that $\dot{\alpha}(s) \neq 0$, $\forall s \in [0, 2\pi]$. Define the velocity vector of the curve as $\dot{\alpha}(s) = e^{\phi(s)} e^{j\theta(s)}$, where $\phi: [0, 2\pi] \rightarrow \mathbb{R}$ and $\theta: [0, 2\pi] \rightarrow \mathbb{R}$ are smooth, and $j = \sqrt{-1}$. The function ϕ is the *speed* of α and measures the rate of stretching and compression, whereas θ is the angle made by $\dot{\alpha}(s)$ with the X -axis and denotes bending. We will represent α via the pair $\nu \equiv (\phi, \theta)$, and denote by \mathcal{H} the collection of all such pairs. In order to make the shape representation invariant to rigid motions and uniform scalings, we restrict shape representatives to pairs (ϕ, θ) satisfying the conditions;

$$\mathcal{C} = \left\{ (\phi, \theta) : \int_0^{2\pi} e^{\phi(t)} dt = 2\pi, \frac{1}{2\pi} \int_0^{2\pi} \theta(t) e^{\phi(t)} dt = \pi, \int_0^{2\pi} e^{\phi(t)} e^{j\theta(t)} dt = 0 \right\} \subset \mathcal{H},$$

where \mathcal{C} is called the *pre-shape space* of planar elastic strings.

Remark: Note that the pair (ϕ, θ) represents the shape of β , and thus ignores its placement, orientation, and scale. Shape deformations are studied using geodesics in the shape space \mathcal{S} connecting them. Given two shapes ν_1 and ν_2 , computing a geodesic involves finding a tangent direction $g \equiv (h, f)$, such that the exponential map [1], $\mathbf{exp}_{\nu_1}(g) = \nu_2$. This is also represented by the geodesic flow $\Psi_1(\nu_1, g) = \nu_2$. Figure 1 shows such a geodesic between two shapes. Shape geodesics are computed under the following Riemannian metric [3]: Given $(\phi, \theta) \in \mathcal{C}$, let h_i and f_i , $i = 1, 2$ be

tangent to \mathcal{C} at (ϕ, θ) . For $a, b > 0$, define

$$\begin{aligned} \langle (h_1, f_1), (h_2, f_2) \rangle_{(\phi, \theta)} &= a \int_0^{2\pi} h_1(s) h_2(s) e^{\phi(s)} ds + \\ &+ b \int_0^{2\pi} f_1(s) f_2(s) e^{\phi(s)} ds. \end{aligned} \quad (1)$$

The parameters a and b control the *tension* and *rigidity* in the metric. The geodesic distance, (used as the shape metric) is now given by

$$d(\nu_1, \nu_2) \triangleq \|(h, f)\|_{(\phi, \theta)} = \sqrt{\langle (h, f), (h, f) \rangle_{(\phi, \theta)}}$$

The remainder of the paper is organized as follows. Section II outlines a clustering algorithm using the geodesic lengths discussed above. The results and the performance of the clustering algorithm are demonstrated in Section III followed by the conclusion.

II. SHAPE CLUSTERING

Classical clustering algorithms on Euclidean spaces generally fall into two main categories: partitional and hierarchical [8]. Assuming that the desired number k of clusters is known, partitional algorithms typically seek to minimize a cost function Q_k associated with a given partition of the data set into k clusters. Usually, the total variance of a clustering is a widely used cost function. Hierarchical algorithms, in turn, take a bottom-up approach. If the data set contains n points, the clustering process is initialized with n clusters, each consisting of a single point. The clusters are then merged successively according to some criterion until the number of clusters is reduced to k . Commonly used metrics include the distance of the means of the clusters, the minimum distance between elements of clusters, and the average distance between elements of the clusters. In this paper, we choose a value of k beforehand.

A. Minimum-Variance Clustering

Consider the problem of clustering n shapes (in \mathcal{S}) into k clusters. To motivate our algorithm, we begin with a discussion of a classical clustering procedure for points in Euclidean spaces, which uses the minimization of the total variance of clusters as a clustering criterion. More precisely, consider a data set with n points $\{y_1, y_2, \dots, y_n\}$ with each $y_i \in \mathbb{R}^d$. If a collection $C = \{C_i, 1 \leq i \leq k\}$ of subsets of \mathbb{R}^d partitions the data into k clusters, the total variance of C is defined by $Q(C) = \sum_{i=1}^k \sum_{y \in C_i} \|y - \mu_i\|^2$, where μ_i is the mean of data points in C_i . The term $\sum_{y \in C_i} \|y - \mu_i\|^2$ can be interpreted as the total variance of the cluster C_i . The total variance is used instead of the (average) variance to avoid placing a bias on large clusters, but when the data is fairly uniformly scattered, the difference is not

significant and either term can be used. The widely used *k-Means Clustering Algorithm* is based on a similar clustering criterion (see e.g. [8]). The *soft k-Means Algorithm* is a variant that uses ideas of simulated annealing to improve convergence [9], [7]. These ideas can be extended to shape clustering using $d(\nu, \mu_i)^2$ instead of $\|y - \mu_i\|^2$, where $d(\cdot, \cdot)$ is the geodesic length and μ_i is the Karcher mean [5] of a cluster C_i on the shape space.

Clustering algorithms that involve finding means of clusters are only meaningful in metric spaces where means can be defined and computed. However, the calculation of Karcher means of large shape clusters is a computationally demanding operation. Therefore, it is desirable to replace quantities involving the calculation of means by approximations that can be derived directly from distances between the corresponding data points. Hence, we propose a variation that replaces $d(\nu, \mu_i)^2$ with the average distance-square $V_i(\nu)$ from ν to elements of C_i . If n_i is the size of C_i , then $V_i(\nu) = \frac{1}{n_i} \sum_{\nu' \in C_i} d(\nu, \nu')^2$. The cost Q associated with a partition C can be expressed as

$$Q(C) = \sum_{i=1}^k \frac{2}{n_i} \left(\sum_{\nu_a \in C_i} \sum_{b < a, \nu_b \in C_i} d(\nu_a, \nu_b)^2 \right). \quad (2)$$

If the average distance-square within the clusters is used, the scale factor in each term is modified to $\frac{2}{n_i(n_i-1)}$. In either case, we seek configurations that minimize Q , i.e., $C^* = \operatorname{argmin} Q(C)$. In this paper we have used the latter cost function.

B. Clustering Algorithm

We will minimize the clustering cost using a Markov chain Monte Carlo (MCMC) search process on the configuration space. The basic idea is to start with a configuration of k clusters and keep on reducing Q by re-arranging shapes amongst the clusters. The re-arrangement is performed in a stochastic fashion using two kinds of moves. These moves are performed with probability proportional to negative exponential of the Q value of the resulting configuration.

- 1) **Move a shape:** Here we select a shape randomly and re-assign it to another cluster. Let $Q_j^{(i)}$ be the clustering cost when a shape ν_j is re-assigned to the cluster C_i keeping all other clusters fixed. If ν_j is not a singleton, i.e. not the only element in its cluster, then the transfer of ν_j to cluster C_i is performed with the probability:

$$P_M(j, i; T) = \frac{\exp(-Q_j^{(i)}/T)}{\sum_{i=1}^k \exp(-Q_j^{(i)}/T)}, \quad i = 1, 2, \dots, k$$

Here T plays the role of temperature as in simulated annealing. Note that moving ν_j to any other cluster

is disallowed if it is a singleton in order to fix the number of clusters at k .

- 2) **Swap two shapes:** Here we select two shapes from two different clusters and swap them. Let $Q^{(1)}$ and $Q^{(2)}$ be the Q -values of the original configuration (before swapping) and the new configuration (after swapping), respectively. Then, swapping is performed with the probability:

$$P_S(T) = \frac{\exp(-Q^{(2)}/T)}{\sum_{i=1}^2 \exp(-Q^{(i)}/T)}.$$

Additional types of moves can also be used to improve the search over the configuration space although their computational cost becomes a factor too. In view of the computational simplicity of moving a shape and swapping two shapes, we have restricted the algorithm to these two simple moves.

In order to seek global optimization, we have adopted a simulated annealing approach. That is, we start with a high value of T and reduce it slowly as the algorithm search for configurations with smaller dispersions. Additionally, the moves are performed according to a Metropolis-Hastings algorithm (see [6] for reference), i.e. candidates are proposed randomly and accepted according to certain probabilities (P_M and P_S above). Although simulated annealing and the random nature of the search help in getting out of local minima, the convergence to a global minimum is difficult to establish. As described in [6], the output of this algorithm is a Markov chain but is neither homogeneous nor convergent to a stationary chain. If the temperature T is decreased slowly, then the chain is guaranteed to converge to a global minimum. However, it is difficult to make an explicit choice of the required rate of decrease in T and instead we rely on empirical studies to justify this algorithm. It is important to note that once the pairwise distances are computed, they are not computed again in the iterations. Secondly, unlike k -mean clustering mean shapes are not used here. These factors make Algorithm 1 efficient and effective in clustering diverse shapes.

We have applied Algorithm 1 to organize a collection of $n = 3270$ shapes (not shown) from the ETH-80 shape database [10] into 25 clusters. Figure 2 shows a few sample images of common objects, and their shape representations from the ETH-80 dataset. Shown in Figure 3(a) are a few samples from the 25 clusters. The elastic metric used in computing pairwise distances for the clusters shown in Fig. 3 assumes the values of $a = b = 1$ in Eqn. 1.

In each run of Algorithm 1, we keep the configuration with minimum Q value. Figure 3(b) shows an evolution of the search process where the Q values are plotted against the iteration index. Figure 3(c) shows a histogram of the

Algorithm 1: For n shapes and k clusters initialize by randomly distributing n shapes among k clusters. Set a high initial temperature T .

- 1) Compute pairwise geodesic distances between all n shapes. This requires $n(n - 1)/2$ geodesic computations.
 - 2) With equal probabilities pick one of two moves:
 - a) **Move a shape:**
 - i) Pick a shape ν_j randomly. If it is not a singleton in its cluster then compute $Q_j^{(i)}$ for all $i = 1, 2, \dots, k$.
 - ii) Compute the probability $P_M(j, i; T)$ for all $i = 1, \dots, k$ and re-assign ν_j to a cluster chosen according to the probability P_M .
 - b) **Swap two shapes:**
 - i) Select two clusters randomly, and select a shape from each of them.
 - ii) Compute the probability $P_S(T)$ and swap the two shapes according to that probability.
 - 3) Update temperature using $T = T/\beta$ and return to Step 2. We have used $\beta = 1.0001$ in our experiments.
-

best Q values obtained in 100 such runs, each starting from a random initial configuration. It must be noted that 80% of these runs result in configurations that are quite close to the optimal. Once pairwise distances are computed, it takes approximately 40 seconds to perform 45,000 steps of Algorithm 1 in the matlab environment. The success of Algorithm 1 in clustering these diverse shapes is visible in these results as similar shapes have been clustered together.

C. Hierarchical Classification

An important goal of this paper is to organize large databases of shapes in a fashion that allows for efficient searches. One way of accomplishing this is by organizing shapes in a tree structure, such that shapes are refined regularly as we move down the tree. In other words, objects are organized (clustered) according to coarser differences (in their shapes) at top levels and finer differences at lower levels. This is accomplished in a bottom up construction as follows: start with all the shapes at the bottom level and cluster them according to Algorithm 1 for a pre-determined k . Then, compute a mean shape for each cluster and at the next level cluster these mean shapes according to Algorithm 1. Applying this idea repeatedly, one obtains a tree organization of shapes in which shapes change from coarse to fine as we move down the tree. Critical to this organization is the notion of the mean of shapes for which we utilize Karcher means.

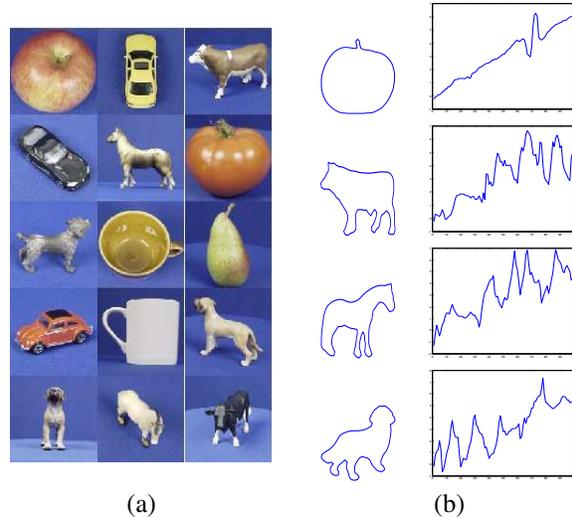


Fig. 2. (a) Examples of images from the ETH-80 dataset. (b) Examples of a few shapes and their angle functions.

We follow the procedure above to generate an example of a tree structure (Fig. 4) obtained for 3270 shapes selected from the ETH-80 database. It is interesting to study the variations in shapes as we follow a path bottom level, these 300 shapes are clustered in $k = 25$ clusters, with the clusters denoted by the indices of their element shapes. Computing the means of each these clusters, we obtain shapes to be clustered at the next level. Repeating the clustering for $k = 8$ clusters we obtain the next level and their mean shapes. In this example, we have chosen to organize shapes in six levels with a single shape at the top. The choice of parameters such as the number of levels, and the number of clusters at each level, depends on the required search speed and performance. It is interesting to study the variations in shapes as we follow a path from top to bottom in this tree. This hierarchical representation of shapes can be effectively used to compare highly dissimilar shapes at a low resolution while allowing similar shapes to be compared at a higher resolution.

III. RETRIEVAL PERFORMANCE AND RESULTS

A logical way to retrieve searches from the hierarchical database is to start at the top, compare the query with the shapes at each level, and proceed down the branch that leads to the best match. At any level of the tree, there is a number, say k , of possible shapes, and our goal is to find the shape that matches the query ν the best. This can be performed using $k - 1$ nearest-neighbor tests leading to the selection of the best hypothesis. In the current implementation, we have assumed a simplification that the covariance matrices for all hypotheses at all levels are identity and only the mean shapes are needed to organize the database. For identity

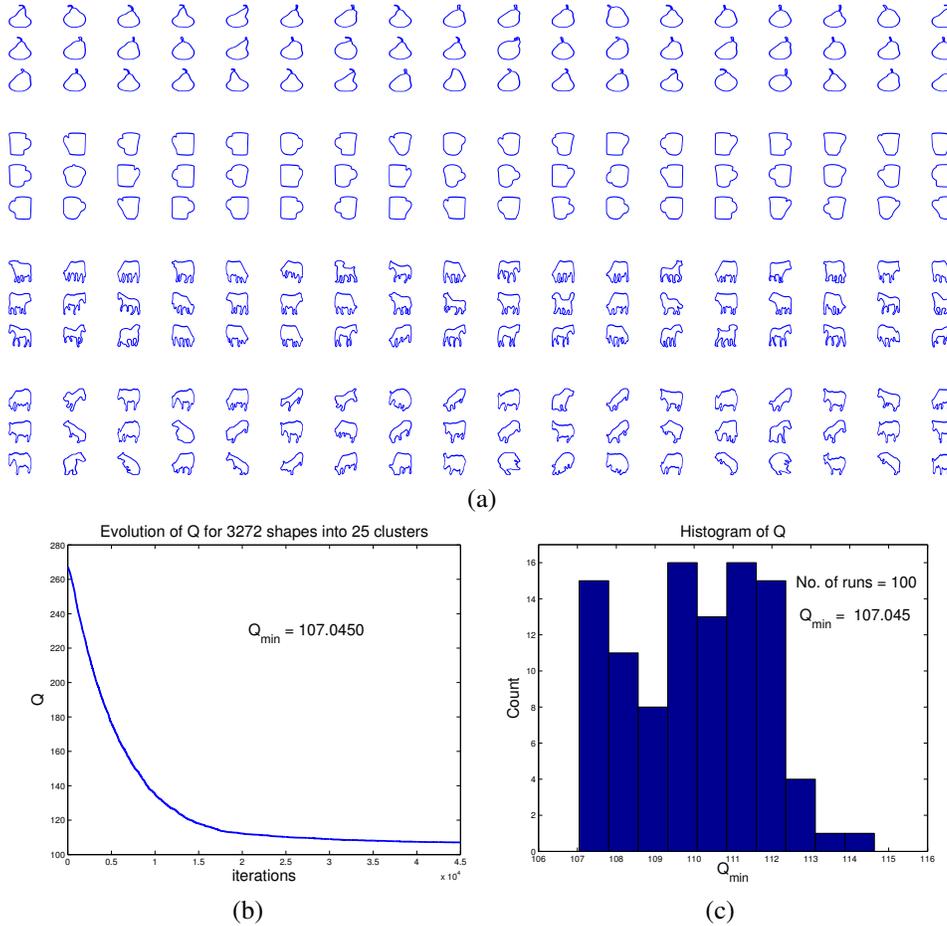


Fig. 3. (a) Examples of shapes from clusters 8,14,16,23 of the ETH-80 database. (b) Sample evolution of Algorithm 1 for the configuration in (a). (c) Histogram of $Q(C^*)$ for 100 runs.

covariances, the task of finding the best match at any level reduces to finding the nearest meanshape at that level. Let μ_i be the given shapes at a level and let x_i be the Fourier vector that encode tangent direction from ν to μ_i . Then, the nearest shape is indexed by $\hat{i} = \operatorname{argmin}_i \|x_i\|$. Proceed down the tree following the nearest shape $\mu_{\hat{i}}$ at each level. This continues until we reach the last level and have found an overall match to the given query. We have implemented this idea using test images from the ETH database. For each test image, we first extract the contour, compute its shape representation as $\nu \in \mathcal{S}$, and follow the tree, shown in Fig. 4, for retrieving similar shapes from the database.

Fig. 5 presents some pictorial examples from this experiment. Shown in the left panels are the original images and in the second left panels their automatically extracted contours. The third column shows five nearest shapes retrieved in response to the query. Finally, the last panel states the time taken for the hierarchical search. In this

experiment, retrieval performance is defined with respect to the original labels, e.g., apple, car, pear, etc. Shown in Fig. 6 are plots of retrieval performances, measured using two different quantities. The first quantity is the precision rate, defined as the ratio of number of relevant shapes retrieved, i.e., shapes from the correct class, to the total number of shapes retrieved. Ideally, this quantity should be one, or quite close to one. The second quantity, called the recall rate, is the ratio of number of relevant shapes retrieved to the total number of shapes in that class in the database. Fig. 6(a) shows average variation of precision rate plotted against the number of shapes retrieved, for four different classes –apple, car, pear, and tomato. As these curves indicate, the retrieval performance of apple falls quickly while that for the other classes remains high. The reason for a low-retrieval performance of apple shapes is their close resemblance in shape to tomatoes. Fig. 6(b) shows plots of recall rate plotted against the number of

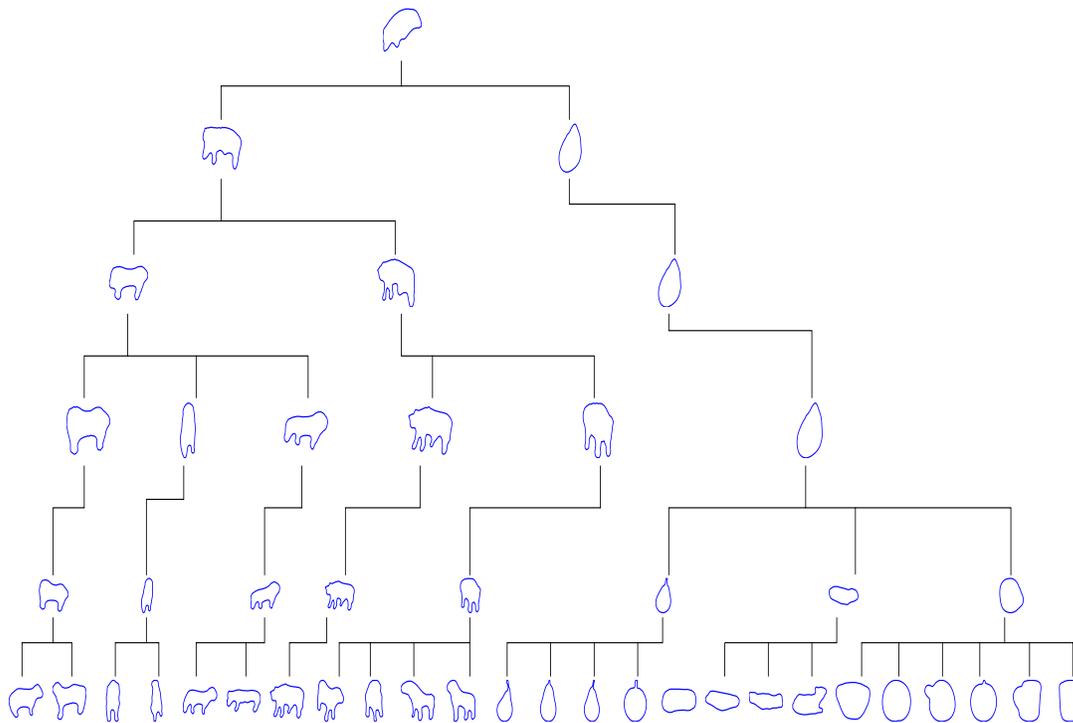


Fig. 4. Hierarchical Organization of 3270 shapes from the ETH-80 database.

shapes retrieved, and Fig. 6(c) plots precision rate against the recall rate, for the same four classes.

IV. CONCLUSION

We have presented a hierarchical organization of shapes based upon an elastic shape-distance metric which utilizes the Riemannian structure of the shape space. Clustering is performed efficiently by minimizing the pair-wise average variance within the clusters and can be used in clustering of shape databases of objects. Hierarchical clustering reduces the search and test times for shape queries against large databases. This has enormous potential for systems which use shape based object retrieval.

REFERENCES

- [1] William M. Boothby. *An Introduction to Differential Manifolds and Riemannian Geometry*. Academic Press, Inc., 1986.
- [2] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(3):372–383, 2004.
- [3] W. Mio and A. Srivastava. Elastic-string models for representation and analysis of planar shapes. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2004.
- [4] A. Srivastava, A. Jain, S. Joshi, and D. Kaziska. Statistical shape models using elastic-string representations. In *Proceedings of Asian Conference on Computer Vision*, 2005.
- [5] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning and testing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(4):590–602, 2005.
- [6] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Text in Stat., 1999.
- [7] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE*, 86(11):2210–2239, November 1998.
- [8] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.
- [9] D. E. Brown and C. L. Huntley. A practical application of simulated annealing to clustering. Technical Report IPC-TR-91-003, Institute for Parallel Computing, University of Virginia, Charlottesville, VA, 1991.
- [10] B. Leibe and B. Schiele. Analyzing Appearance and Contour Based Methods for Object Categorization. *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2003.
- [11] A. Vailaya and M. Figueiredo and A. Jain and H. Zhang. Image Classification for Content-Based Indexing. *IEEE Transactions on Image Processing*, 10(1):117-130, 2001.
- [12] A. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, August 1996.
- [13] F. Mokhtarian, S. Abbasi and J. Kittler. Efficient and robust shape retrieval by shape content through curvature scale space. *Proceedings of First International Conference on Image Database and MultiSearch*, 1996.
- [14] F. Mokhtarian and A. Mackworth A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(8):789–805, 1992.
- [15] Sharvit, D. and Chan, J. and Tek, H. and Kimia, B.B. Symmetry-

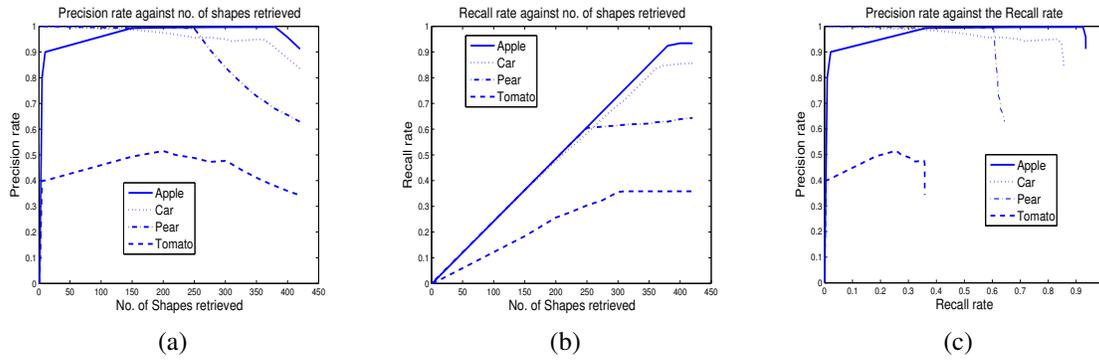


Fig. 6. (a) Precision rate versus number of shapes retrieved, (b) recall rate versus number retrieved, and (c) precision rate versus recall rate.

Shape Image	Shape Contour	Retrieved shapes from hierarchy

Fig. 5. Examples of shape retrieval using hierarchical organization.

based indexing of image databases. *Content-Based Access of Image and Video Libraries*, 1998

- [16] G. Cortelazzo and G. A. Mian and G. Vezzi and P. Zamperoni. Trademark shapes description by string-matching techniques. *Pattern Recognition*, 27(8):1005–1018,1994.
- [17] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. Computer*, C-21, 269–281, 1972.
- [18] E. Persoon and K. S. Fu. Shape discrimination using Fourier descriptors. *IEEE Trans. Systems, Man, and Cybernetics*,(7):170–179, Mar. 1977.
- [19] J. Gary and R. Mehrotra. Shape similarity-based retrieval in image database systems. *In Proc. of SPIE*, 1662:2–8, 1992.
- [20] D. G Shen and Horace. H. S Ip. Discriminative Wavelet shape Descriptors for Invariant recognition of 2D patterns, *Pattern Recognition*, 32(2):151–166, February, 1999.
- [21] M. R. Teague. Image analysis via the general theory of moments. *J. Optical Soc. of America*,70(8):920–930, 1980.